Functional Requirements:

Visitor:

- Signup
- login
- Browse and search Teachers
- Browse and search courses
- Browse teacher profile
- Browse course profile

User:

- Manage Profile(Edit)
- Change password
- Manage Favorates(add, delete, browse favorates)
- Manage private Masseges(send to user, receive message, browse messages log, browse messages log with specific user)
- Report(teacher, course,review).
-

Teacher:

- Manage his Courses(add, edit, delete, browse my courses)
- Withdraw from balance.

Student:

- Send message to teacher.
- Enroll to course
- Cancel enrollment
- Deposit funds.
- Browse course info.
- Browse course details(lessons)

Admin:

- Manage Categories
- Manage school subjects
- Manage universities subjects
- Manage courses(approve, reject)
- Manage Reports(browse reports, display report info, approve, confirm or reject with comment)
-

Non-functional Requirements:

1) Very speed response on google pagespeed tool:
   a) Page of search teachers: average optimization >=60/100
   b) Page of search courses: average optimization >=60/100
   c) Welcome page: average optimization >=80/100
2) Automatic backup(every day)
3) User friendly: we will provide you with the style.
4) Used technologoies: MVC, using codeigniter,using bootstrap for styling.
5) Mobile responsive pages.

Prototypes:

Main Scenario:

How system works:

1. Teacher signup and enter his details
2. Student signup.
3. Student login.
4. Student searchs for teacher by specific category or university subject or school subject.
5. Student send message to nominated teachers.
6. Teachers responds to messages.
7. Student choose teacher and agree with him on fee, duration and date/time and of the lesson.
8. Student send lesson request to choosed teacher.
9. Teacher receives lesson request, if he approves then a lesson will be created, else if he rejected then the lesson request will be finished. Else he can put a comment on the lesson request to let the student modify lesson request and resend it to him another time
10. Teacher and student attend lesson on time and start lesson.
11. Teacher finished lesson in coordination with student.
12. Student put review on lesson or submit a complaint.
13. If student submit a complaint then a dispute will be created automatically.
14. Student write his complaint details and put his request to resolve dispute (refund all fees, or refund a percentage from fees).
15. Teacher receives dispute info and write his response on student requirements and then put his offer to resolve dispute.
16. Teacher and student put final offers, then either resolve it or request judgment from khsoosi judge.
17. Khsoosi judge employee check all lesson and dispute details and decide what to do.

Usecase description:

Usecase Id	1
Usecase Name	Browse course profile
Stackholders	Visitor
Precondition	Student browse courses
Post condition	Course has been displayed to visitor
Main scenario	System displays course info: overview, details, instructor profile link, method of teaching, course image, course level, rates, views number, number of students, and categories of the course.

Usecase Id	2
Usecase Name	Browse course details(lessons)
Stackholders	Student
Precondition	student enroll the course
Post condition	Course lessons has been displayed
Main scenario	Student click on each lesson's link. System displays lesson's details.

Usecase Id	3
Usecase Name	Enroll to course
Stakeholders	Student
Precondition	Student browse the course
Post condition	Student enrolled the course
Main scenario	1. Student click on "enroll to course" 2. System displays the courses fees. 3. Student confirm enrollment. 4. System checks the student's balance, If balance is enough then continue to step 5, Else then <extends> to usecase "deposit funds" and then redo step 4. 5. System enrolls the student to course.

Usecase Id	4
Usecase Name	Cancel enrollment
Stakeholders	Student
Precondition	Student is already enrolled in course

Post condition	Enrollment in course is cancelled
Main scenario	6. Student clicks on cancel button. 7. System display confirmation button. 8. If student not confirmed then finish the usecase, Else then system remove the enrollment of student in the course and display success message.

Usecase Id	5
Usecase Name	request for lesson
Stakeholders	Student, teacher
Precondition	Student browses teacher’s profile
Post condition	Either a lessons profile is created or teacher rejected the request for lesson.
Main scenario	1. Student clicks on “request” button. 2. Start Loop<until request is approved >: Student enters his offer: price of lesson, date and time of the lesson. 3. System sends request notification to teacher. 4. System displays the request info to teacher. 5. Teacher checks the request info, If he accepted then system create a lesson profile with status: “initiated”. 6. System sends notification to student. End loop<when request is approved>
Exceptional scenario	If teacher rejected request then: 1. system asks for reasons of rejection. 2. teacher writes reasons of rejection and send it. 3. system sends notification to student with reason of rejection

Usecase Id	6
Usecase Name	attend lesson
stakeholders	User
Precondition	Lesson profile has been created
Post condition	Lesson profile is finished
Main scenario	1. teacher and student enters the lesson profile page, then press the button “request to start”, and the other user confirm to start.

	2. System displays a video conference page to both teacher and student and turn on the mic and cam. 3. Teacher and student start lesson, when one of them wants to go out from the conference room he clicks on “X” sign, then system returns him to lesson profile page again, teacher and student can return to conference room another time. 4. When lesson if finished both users clicks on “X” of the conference room then return to lesson profile page. 5. Teacher clicks on “finish lesson”. 6. System sends “finish lesson request” to student. 7. Student clicks on “confirm finish lesson”. 8. System closes the lesson profile

Usecase Id	7
Usecase Name	Deposit funds
Stakeholders	Student
Precondition	Balance is low
Post condition	Balance is increased
Main scenario	System displays the current balance and ask student to write the amount of money to deposit. Student writes the amount of money to deposit. System forward student request to the payment gateway.

Usecase Id	8
Usecase Name	Rate lesson
Stakeholders	Student
Precondition	Student attended lesson
Post condition	Lesson is rated
Main scenario	1. System displays lesson’s info and asks student to rate. 2. Student submit his rate and comment. 3. system sends notification to teacher.

Usecase Id	9
Usecase Name	Manage private Messages (send to user, receive message, browse messages log, browse messages log with specific user)
Stakeholders	User
Precondition	User is logged in
Post condition	Messages are managed
Main scenario	Manage private messages includes 4 scenarios: 1) send to user: i) user puts his message and clicks on “send” button. ii) System sends message to receiver and sends him a notification. 2) Receive message: a) User displays message. 3) Browse message log: a) System displays a table containing messages log. 4) Browse messages log with specific user: a) System displays messages log with a user.

Usecase Id	10
Usecase Name	Withdraw from balance.
Stakeholders	Teacher
Precondition	Teacher is logged in
Post condition	Teacher withdraw money
Main scenario	1. System displays balance info. 2. Teacher clicks on “withdraw” button. 3. System asks teacher to enter the money to withdraw(not more than the balance itself). 4. Teacher puts the amount of money to withdraw. 5. System send request to payment gateway and receives feedback. 6. System displays feedback to teacher.

Usecase Id	11

Usecase Name	Report(teacher, course,review)
Stakeholders	User
Precondition	User is logged in
Post condition	Report is submitted
Main scenario	1. User press on “Report” button. 2. System asks user to enter infringement type and report details. 3. User enters infringement type and report details and submit report. 4. System send report to admin.

Usecase Id	12
Usecase Name	Manage Favorates(add, edit, delete, browse favorates)
Stakeholders	User
Precondition	User is logged in
Post condition	Favorates is being managed(added, deleted, browsed)
Main scenario	Add favorate: 1. User presses on the “add to favorates” button on the course/teacher. 2. System adds course/teacher to the user’s favorates and displays success message. delete favorate: 1. User presses on the “remove favorate” button on the favorate course/teacher. 2. System removes course/teacher from user’s favorates and displays success message. Browse favorates: 1. User asks to display favorates screen. 2. System displays favorate courses/teachers screen.

Usecase Id	13
Usecase Name	Manage his Courses(add, edit, delete, browse my courses)
Stakeholders	Teacher, admin
Precondition	teacher is logged in, admin is logged in
Post condition	Teacher’s courses is being managed(added, deleted, browsed)
Main scenario	Add course: 3. teacher presses on the “create course” button. 4. System asks teacher to enter course details and lessons. 5. Teacher enters his course details and submit it. 6. System send course add request to admin.

	7. Admin displays course details and takes decision approve or reject and write comment if he has. 8. System sends notification to teacher. Edit course: 1. Teacher opens course details for edit. 2. Teacher edit course details(title, about, details, lessons, image). 3. System sends a notification to admin. 4. Admin view course details and review it, and takes decision approve or reject and write comment if he has. Delete course: 3. teacher presses on the “delete” button on the course details page. 4. System display a confirmation message. 5. If teacher confirmed then continue to next step. 6. System sends deletion request to admin. 7. Admin review course info and check number of current enrolled students, if there are a current enrollments then admin rejects the deletion request, else then admin deletes course(deleted course will not be deleted from database, and will still be displayed for the previous enrolled students but with a warning sign saying that its deleted). Browse my courses: 3. teacher asks to display his courses. 4. System displays teacher’s courses screen.

Usecase Id	14
Usecase Name	Browse and search Teachers
Stakeholders	visitor
Precondition	-
Post condition	Search result is displayed
Main scenario	1. visitor enters search criteria(either category, or university subject with university name, or school curriculums and grade). 2. System displays search result.

Usecase Id	15
Usecase Name	Browse and search courses
Stakeholders	Visitor
Precondition	-
Post condition	Search result is displayed
Main scenario	1. visitor enters search criteria(either category, or university subject with university name, or school curriculums and grade).

	2. System displays search result.